SMC-TR-05-20

AEROSPACE REPORT NO.
TR-2005(8550)-1

Software Acquisition Best Practices Tutorial

30 September 2005

Prepared by

S. ESLINGER and M. A. RICH
Software Engineering Subdivision
Computer Systems Division

R. J. ADAMS, and K. L. OWENS
Software Acquisition and Process Department
Software Engineering Subdivision

Prepared for

SPACE AND MISSILE SYSTEMS CENTER
AIR FORCE SPACE COMMAND
2430 E. El Segundo Boulevard
Los Angeles Air Force Base, CA 90245

Engineering and Technology Group

THE AEROSPACE CORPORATION
El Segundo, California

This report was submitted by The Aerospace Corporation, El Segundo, CA 90245-4691, under Contract No. FA8802-04-C-0001 with the Space and Missile Systems Center, 2430 E. El Segundo Blvd., Los Angeles Air Force Base, CA 90245. It was reviewed and approved for The Aerospace Corporation by Mary A. Rich, Principal Director, Software Engineering Subdivision, Computer Systems Division. Michael Zambrana was the project officer for the Mission-Oriented Investigation and Experimentation (MOIE) program.

This report has been reviewed by the Public Affairs Office (PAS) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nationals.

This technical report has been reviewed and is approved for publication. Publication of this report does not constitute Air Force approval of the report's findings or conclusions. It is published only for the exchange and stimulation of ideas.

Michael Zambrana
SMC/AXE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE (*DD-MM-YYYY*)	2. REPORT TYPE	3. DATES COVERED *(From - To)*
30/09/05		

4. TITLE AND SUBTITLE

Software Acquisition Best Practices

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

S. Eslinger, R. J. Adams, K. L. Owens, and M. A. Rich

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

The Aerospace Corporation
El Segundo, CA 90245-4691

8. PERFORMING ORGANIZATION REPORT NUMBER

TR-2005(8550)-1

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Space and Missile Systems Center
Air Force Space Command
2450 E. El Segundo Blvd.
Los Angeles Air Force Base, CA 90245

10. SPONSOR/MONITOR'S ACRONYM(S)

SMC

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

SMC-TR-05-20

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This tutorial presents a comprehensive set of software acquisition best practices that have been defined based on experience with numerous software-intensive space system acquisitions. The tutorial covers these software acquisition best practices in chronological order through each phase of the National Security Space acquisition life cycle. It addresses important software acquisition best practices that must be carried out by program offices throughout the acquisition life cycle, from program inception through retirement. This tutorial was presented at the Space Systems Engineering and Risk Management Symposium in October 2005.

15. SUBJECT TERMS

software; software acquisition; software engineering; best practices

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Suellen Eslinger
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED		84	19b. TELEPHONE NUMBER *(include area code)* (310)336-2906

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. 239.18

Acknowledgements

Development of the *Software Acquisition Best Practices Tutorial* and preparation of this report was funded by the Software Acquisition task of the Mission-Oriented Investigative Experimentation (MOIE) research project.

Foreword

Proposed changes in acquisition policy over the past few years have recently been solidified with the establishment of new versions of DODD 5000.1 and DODI 5000.2, a new capability-based process for requirements generation, and new acquisition policy specific to National Security Space (NSS) systems (NSS 03-01). Many mandates requiring use of acquisition reform practices defined during the 1990s have been lifted or modified (e.g., prohibitions against use of military standards, restrictions on quantity and type of technical deliverable documentation, contractor Total System Performance Responsibility). When these changes began to be implemented, programs supported by The Aerospace Corporation began seeking guidance on the "right" course of action for acquiring complex, software-intensive systems. This situation provided the opportunity for identifying and implementing a comprehensive set of software acquisition best practices as part of this new acquisition environment.

Software acquisition best practices, by definition, are practices that have been identified through experience as being significant contributors to the successful acquisition of software-intensive systems. A comprehensive set of best practices must provide a consistent and integrated approach to software acquisition throughout the acquisition life cycle, both pre- and post-contract award. In addition, because software always exists within the context of the system, the software acquisition best practices must be consistent and integrated with a comprehensive set of system acquisition best practices. Finally, the set of best practices must be suitable for acquiring today's complex software systems that will be developed using the latest software development process and product technologies.

A comprehensive set of software acquisition best practices has been defined, based on experience with numerous software-intensive space system acquisitions over 20 years, as part of the Aerospace Mission-Oriented Investigation and Experimentation (MOIE) research effort (Software Acquisition Task). This tutorial covers these software acquisition best practices in chronological order through each phase of the NSS acquisition life cycle. It addresses important software acquisition best practices that must be carried out by program offices throughout the acquisition life cycle, from program inception through retirement. Practical information is provided to enable the immediate application of the best practices to programs. Other sources of best practices from the DOD community are also discussed.

The target audience for this tutorial is anyone responsible for acquiring, or supporting the acquisition of, space systems containing complex, mission-critical software. Note that the tutorial is not targeted solely to software specialists. Also, while the authors developed these software acquisition best practices while working in the space systems domain, the best practices can be applied to all large, complex software-intensive systems in domains requiring high reliability and integrity.

This tutorial has been presented in numerous forums. The version of the tutorial in this technical report is that presented at the Space Systems Engineering and Risk Management Symposium in October 2005.

Contents

Outline

- **Definitions and Background**
  - Software Acquisition vs. Software Engineering
  - Best Practices
- **Software Acquisition Best Practices**
  - Pre-Phase A
  - Phase A
  - Phases B/C/D
  - Full Acquisition Life Cycle
- **Software Acquisition Best Practices from Other Organizations**
  - Data and Analysis Center for Software (DACS)
  - Software Engineering Institute (SEI)
- **Current Air Force and SMC Software Acquisition Policy**
- **Conclusion**
- **Acronym List**

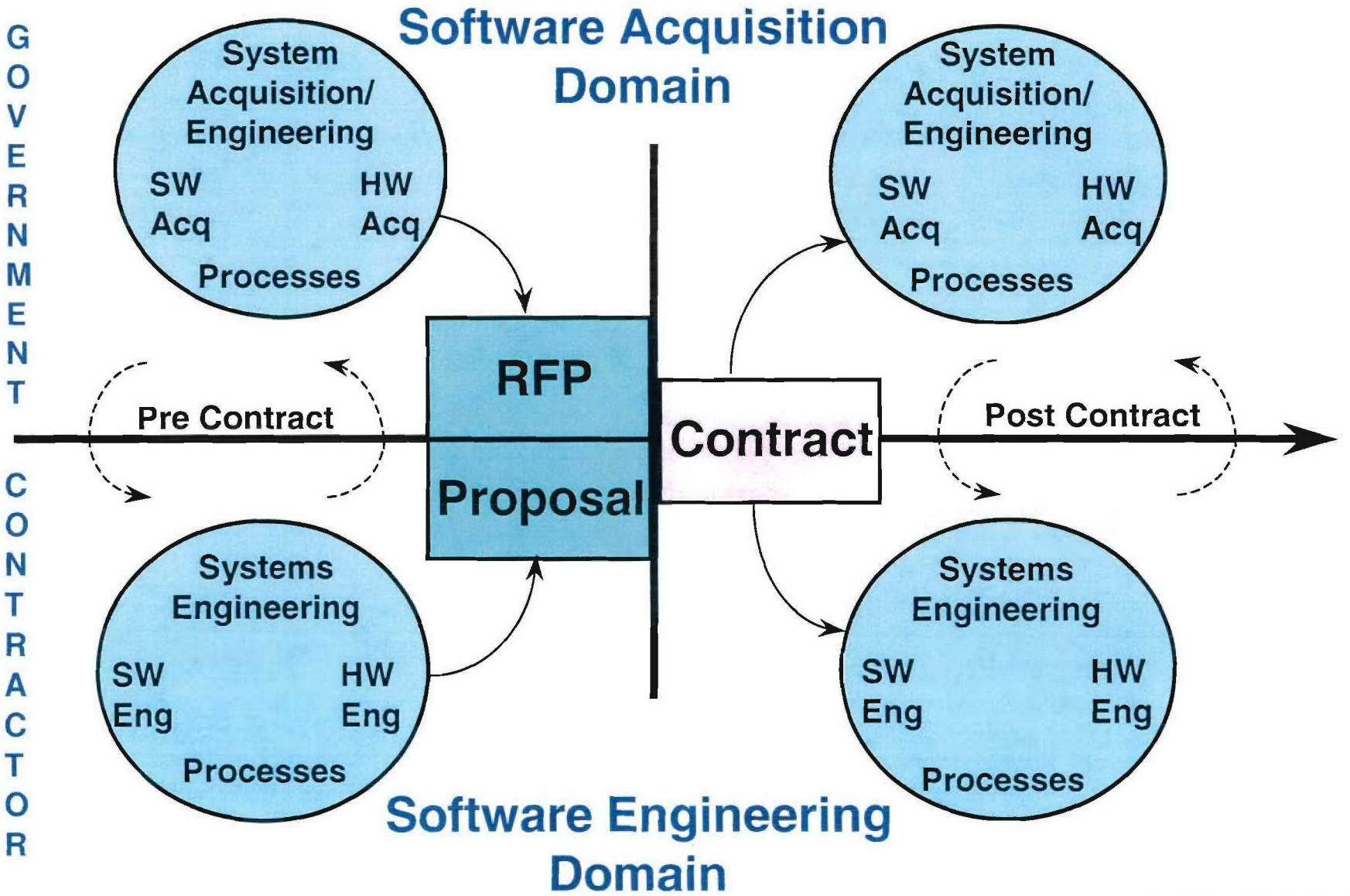
This page is intentionally blank

Software Acquisition vs. Software Engineering

- **Software Acquisition**
  - The set of processes (i.e., the set of policies, procedures, methodologies, tools, etc.) used by the government to acquire software
- **Software Engineering**
  - The set of processes (i.e., the set of policies, procedures, methodologies, tools, etc.) used by the developer to build software

An understanding of software acquisition requires a thorough understanding of software engineering.

Software Acquisition vs. Software Engineering



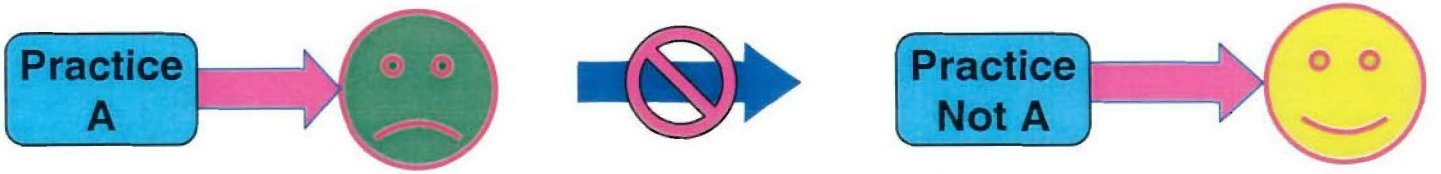
Why Worry About Software Acquisition Processes?

- **Clearly, a successful software development project is dependent on the software engineering processes used**
  - ❖ "The quality of a software product is largely determined by the quality of the process used to develop and maintain it."*
- **However, the software acquisition processes are also highly influential in achieving a successful software development project**
- **The software acquisition processes used can positively encourage, or adversely constrain, the developers in their application of high-quality software engineering processes**

* Paulk, M., et al, *The Capability Maturity Model for Software: Guidelines for Improving the Software Process*, Addison-Wesley, 1994, p. 8.

Best Practices

- **Definition: Best Practices are practices that people with recognized expertise in the subject area have identified through experience as being significant contributors to project success**
- **Negative experience or positive experience may identify Best Practices**
  - However, one must not be trapped by logical fallacies



- **Note that Best Practices (both individually and collectively)**
  - Have not necessarily undergone detailed study
  - Have almost never been analytically determined to be "best"
  - Never form an exhaustive set (there is always the possibility of more)
  - Are not static (they change with new experiences and new technologies)
  - Are dependent on the context and environment

Software Acquisition Best Practices

- **Software Acquisition Best Practices are, therefore, practices that people with recognized software acquisition expertise have identified through experience as being significant contributors to the successful acquisition of software-intensive systems**

- **The Software Acquisition Best Practices presented have been derived from the Aerospace Software Acquisition research team's collective experience in the acquisition of software-intensive space systems**
  - Over 60 collective years of software acquisition experience spanning over 20 years duration
  - Many additional years of experience in developing software, managing software development projects, and leading software process improvement efforts

Characteristics of Space Systems

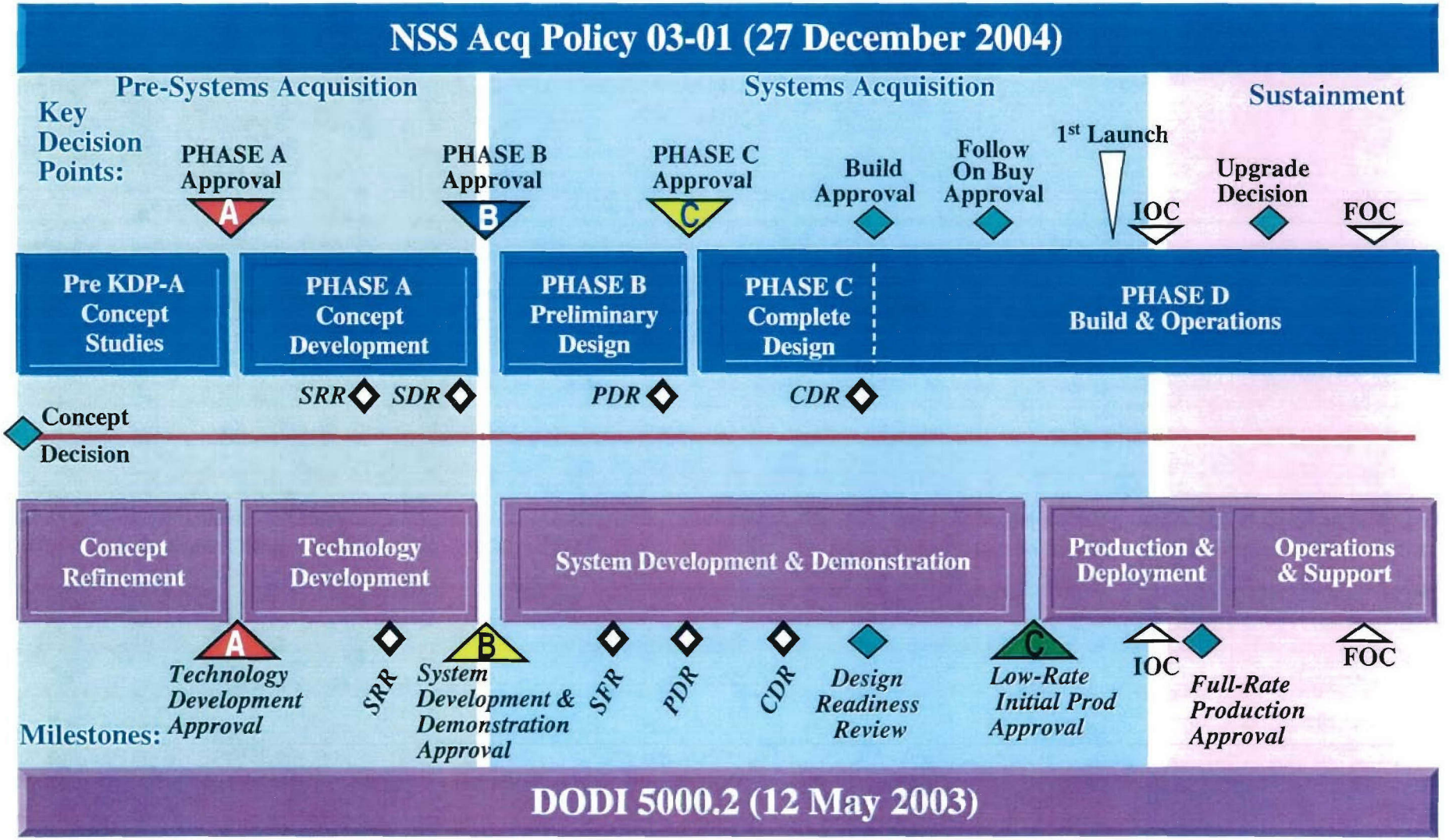
- **Large software-intensive systems**
  - SLOC order of magnitude: 10^5 onboard and $10^6 - 10^7$ on the ground
  - Multi-satellite constellations
  - Multiple ground elements, frequently worldwide
- **Complex combinations of hardware and software**
- **Complex external and internal interfaces**
- **Usually unprecedented**
- **High reliability and integrity requirements**
- **Developed by large teams of multiple contractors**

Software Acquisition Best Practices must support these characteristics.

Additional Characteristics Required of Software Acquisition Best Practices

- **Comprehensive, covering the full system life cycle**
  - Both pre-and post-contract award
- **Consistent and integrated approach**
  - Throughout the life cycle
- **Part of an integrated approach to system acquisition**
  - Software always exists within the context of the system
- **Suitable for acquiring today's complex software systems**
  - Being developed with the latest software development process and product technologies
- **Consistent with the applicable system and software acquisition policies and regulations**
  - DOD level, Service/Component level, Center/Organizational level, etc.

DOD and NSS Acquisition Models*



* From National Security Space Acquisition Policy #03-01, 27 December 2004.

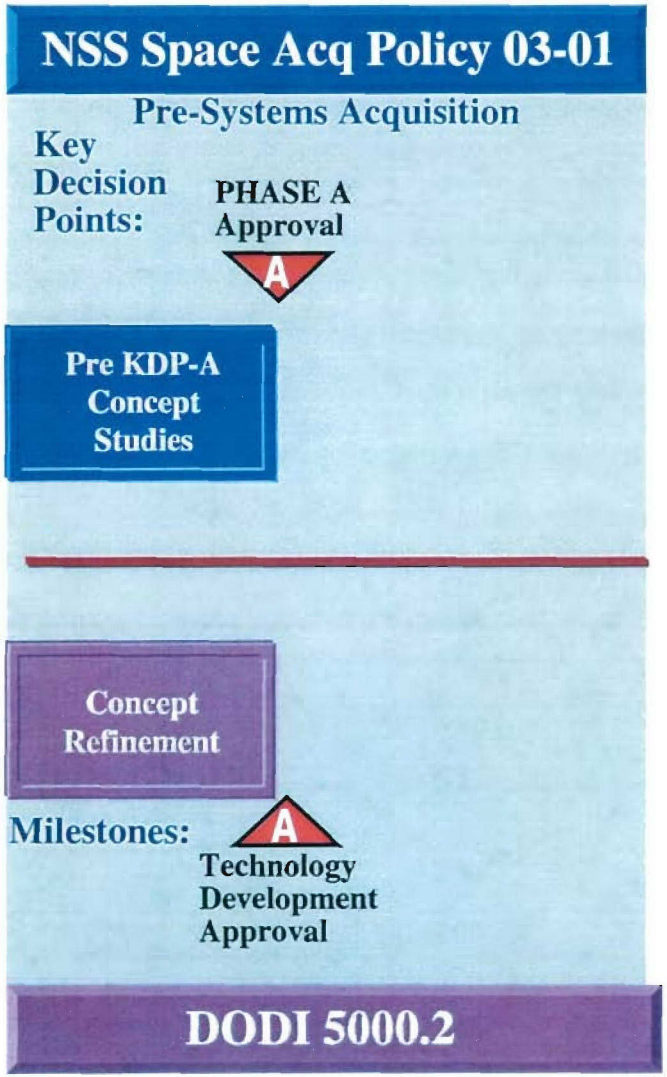
Outline

- **Definitions and Background**
  - Software Acquisition vs. Software Engineering
  - Best Practices
- **Software Acquisition Best Practices**
  - Pre-Phase A
  - Phase A
  - Phases B/C/D
  - Full Acquisition Life Cycle
- **Software Acquisition Best Practices from Other Organizations**
  - Data and Analysis Center for Software (DACS)
  - Software Engineering Institute (SEI)
- **Current Air Force and SMC Software Acquisition Policy**
- **Conclusion**
- **Acronym List**

This page is intentionally blank

Pre-Key Decision Point (KDP) A

Pre-KDP A Best Practices (Concept Studies)



Defining:

- Program life cycle
- Initial system definition
- Initial government architecture concepts
- Initial government cost and schedule baselines
- Executable program evolutions
- Global acquisition and Test and Evaluation (T&E) strategies

Best Practices for Defining the Program Life Cycle

Use a software-friendly acquisition model

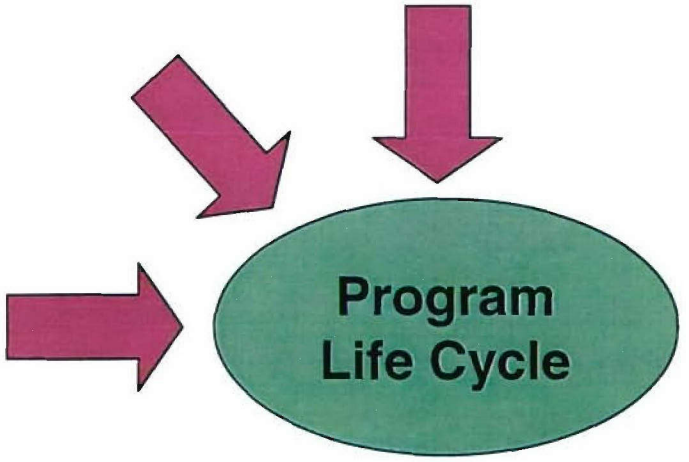
- Evolutionary acquisition is more suited to large, complex software-intensive systems

Tailor the acquisition model for a software-intensive system

- Selection of a single contractor at appropriate point in software development life cycle
- With or without production phase

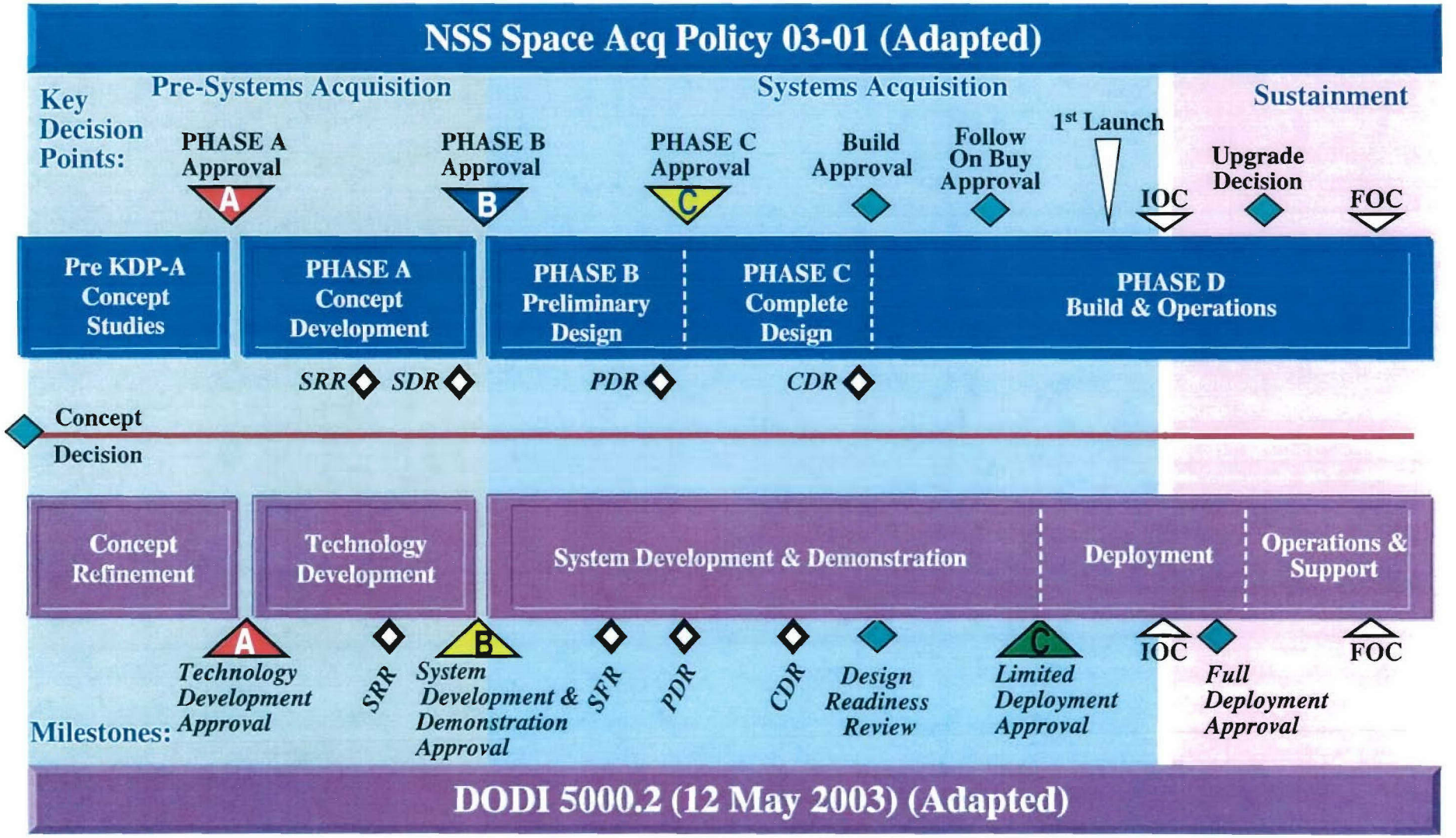
Choose software-friendly points in the life cycle for contract actions

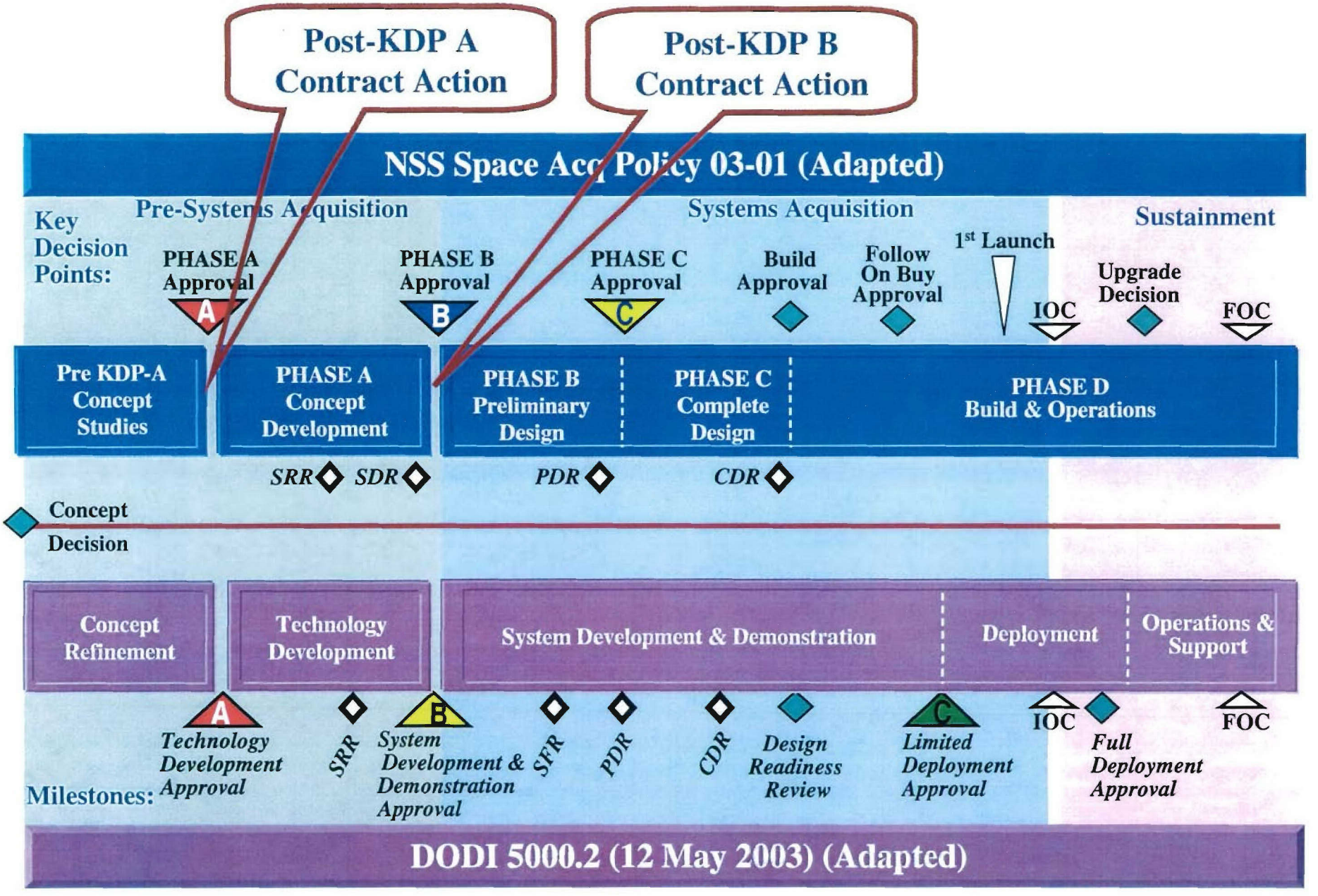
- Avoid contract actions in the middle of software development spirals (e.g., post System PDR)
- Develop firm basis for software costing before MS B/KDP B



Example DOD and NSS Acquisition Models

Tailored for Software-Intensive Systems without Production





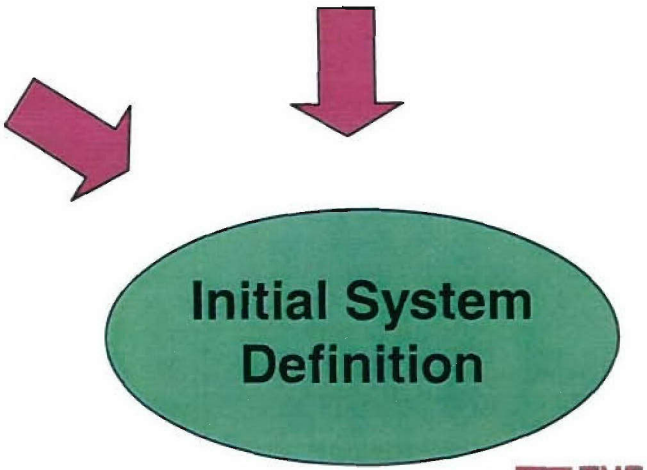
Best Practices for Developing the Initial System Definition

Ensure initial capabilities appropriately include software

- Iterate with user to ensure software-related capabilities are appropriately included in the Initial Capabilities Document (ICD), e.g.,
  - Key Performance Parameters
  - RMA, safety, security (information assurance), human systems integration
  - Interoperability
  - Architecture views

Ensure concept of operations appropriately includes software

- Iterate with user to ensure the feasibility of the user's Concept of Operations (CONOPS) from a software perspective
  - The CONOPS is embodied in the software!



Best Practices for Developing the Initial Government Architecture Concepts

Perform software-inclusive architecture trade studies

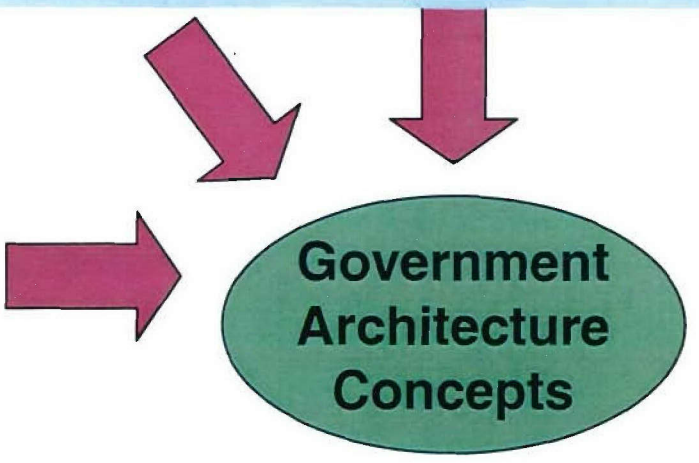
- With system architecture trades
- Identify and address critical HW/SW architecture issues
- Include major legacy components and COTS software

Select a set of integrated HW/SW architecture concepts

- Able to grow with each successive evolution with little expected rework
- Able to integrate each successive evolution with previous evolutions (and legacy system, as applicable)

Include software in evaluation of architecture concepts

- Evaluate software evolution and growth capability
- Include software in life cycle cost analysis (COTS software refresh, legacy and new software re-engineering and maintenance)



Best Practices for Developing the Initial Government Cost and Schedule Baseline

Determine realistic SW size estimates for each evolution

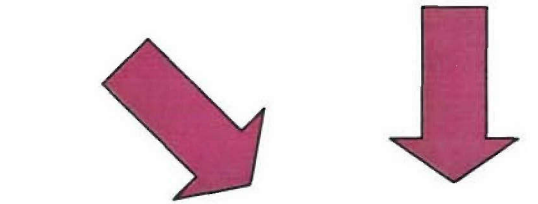
- Use gov't. HW/SW architecture concept
- Include all SW functionality and infrastructure needed
- Use historical data from similar past programs & early concept study data

Determine realistic SW schedule estimates for each evolution

- Include all software effort in schedule
- Never compress software schedule >20% off nominal*

Determine realistic SW effort & cost estimates for each evolution

- Include COTS, reuse & new software
- Include tasks not reflected in cost models (e.g., integration of SW components costed separately, COTS)
- Estimate at least at 80% likelihood level



Government Cost And Schedule Baseline

* Software Program Manager's Network, The Program Manager's Guide to Software Acquisition Best Practices, Version 2.31, p. 22.

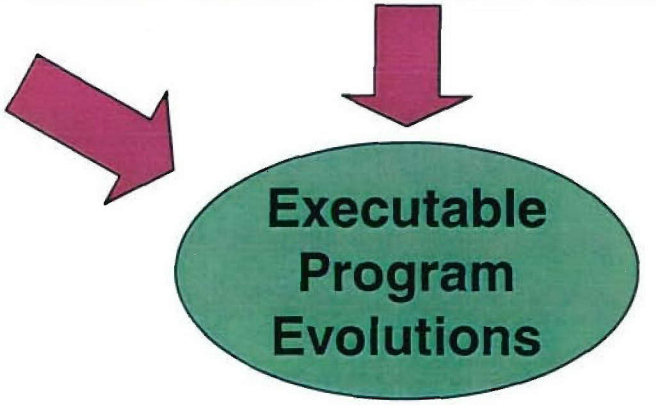
Best Practices for Defining Executable Program Evolutions

Consider SW implications when defining evolution capabilities

- Analyze feasibility of developing the required software for each evolution
  - Based on realistic software size, effort, cost & schedule estimates
  - Include software cost and schedule estimation risk
- Analyze feasibility of integrating the software in each evolution with all previous evolutions (and legacy system(s), as applicable)
  - Based on integrated hardware/ software architecture
- Analyze impacts of COTS software refresh and legacy software upgrades

Consider SW implications when defining evolution schedules

- Analyze feasibility of overlapping software development schedules for closely spaced evolutions
- Avoid plans that require developing subsequent evolutions on unknown software baselines
- Analyze feasibility of COTS refresh and legacy SW upgrade schedules



Best Practices for Developing the Global Acquisition and T&E Strategies

Develop plans for computer system technology insertion

- Include COTS HW and SW refresh in each successive evolution
- Understand new computer HW and SW technologies needed for each evolution and study their readiness

Develop plans for evaluation of contractor software capability

- Perform a government evaluation of contractor team software capability
- Prior to or part of selection of a single development contractor

Develop plans for software support

- Plan for managing multiple baselines (operations and development)
- Plan for integrating software maintenance actions on operational evolutions into evolutions under development

Develop a software-inclusive T&E strategy

- Part of the program's T&E strategy

Software Capability Appraisals - 1

There are two methods for software capability appraisals currently in use

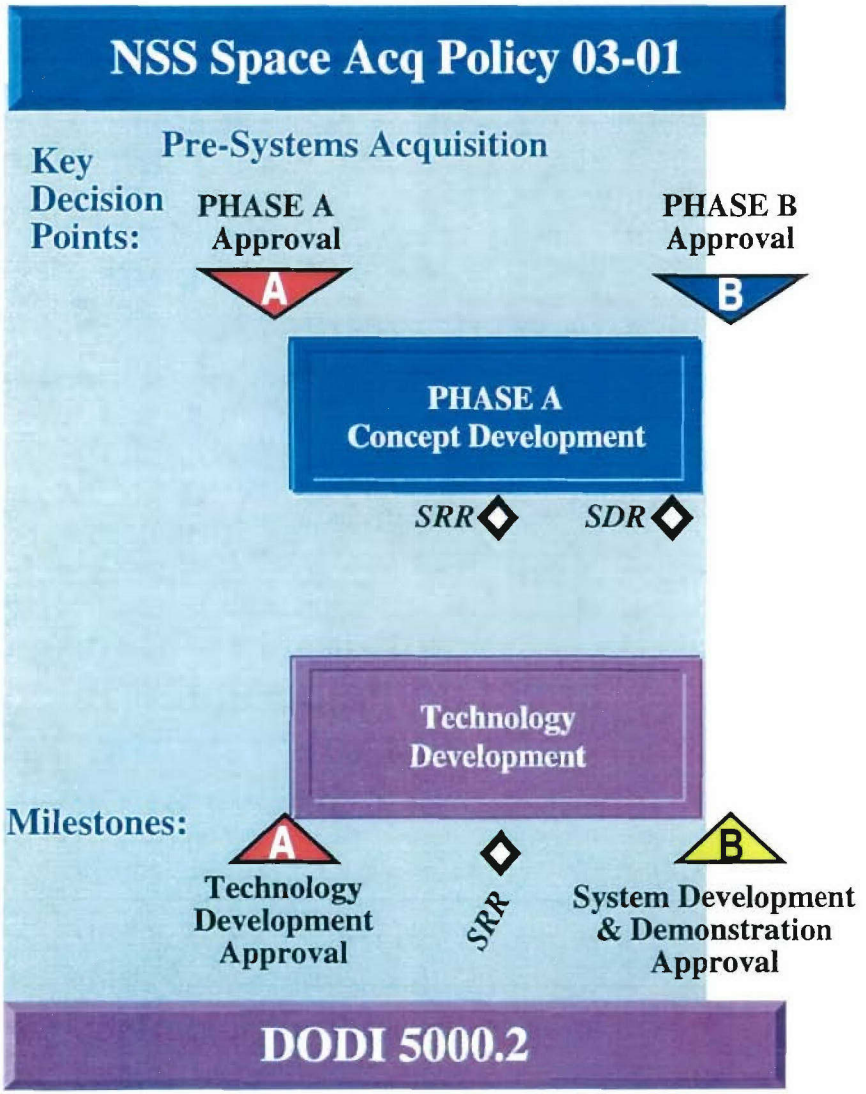
- The Air Force's Software Development Capability Evaluation (SDCE)
  - Air Force Space Command, *Software Development Capability Evaluation,* Volumes 1 and 2, AFMCP 63-103, 15 June 1994
- The SEI's Standard CMMI®-Based Appraisal Method for Process Improvement (SCAMPI℠)
  - Software Engineering Institute. *Standard CMMI® Appraisal Method for Process Improvement (SCAMPI℠), Version 1.1: Method Definition Document*, CMU/SEI-2001-HB-001, December 2001
  - Available from SEI website: http://www.sei.cmu.edu/publications/program/index.html

Software Capability Appraisals - 2

- **Current Recommendations**
  - Perform a SCAMPI[SM] Class C appraisal for source selections
    - Based on answers to questionnaire and documented evidence
    - No site visits required
  - Perform periodic SCAMPI[SM] Class B appraisals for contract monitoring
    - Based on interviews and documented evidence
    - Requires site visits
    - Focused only the program under contract
  - Can do combined software and systems engineering appraisals

Phase A

Phase A Best Practices (Concept Development)

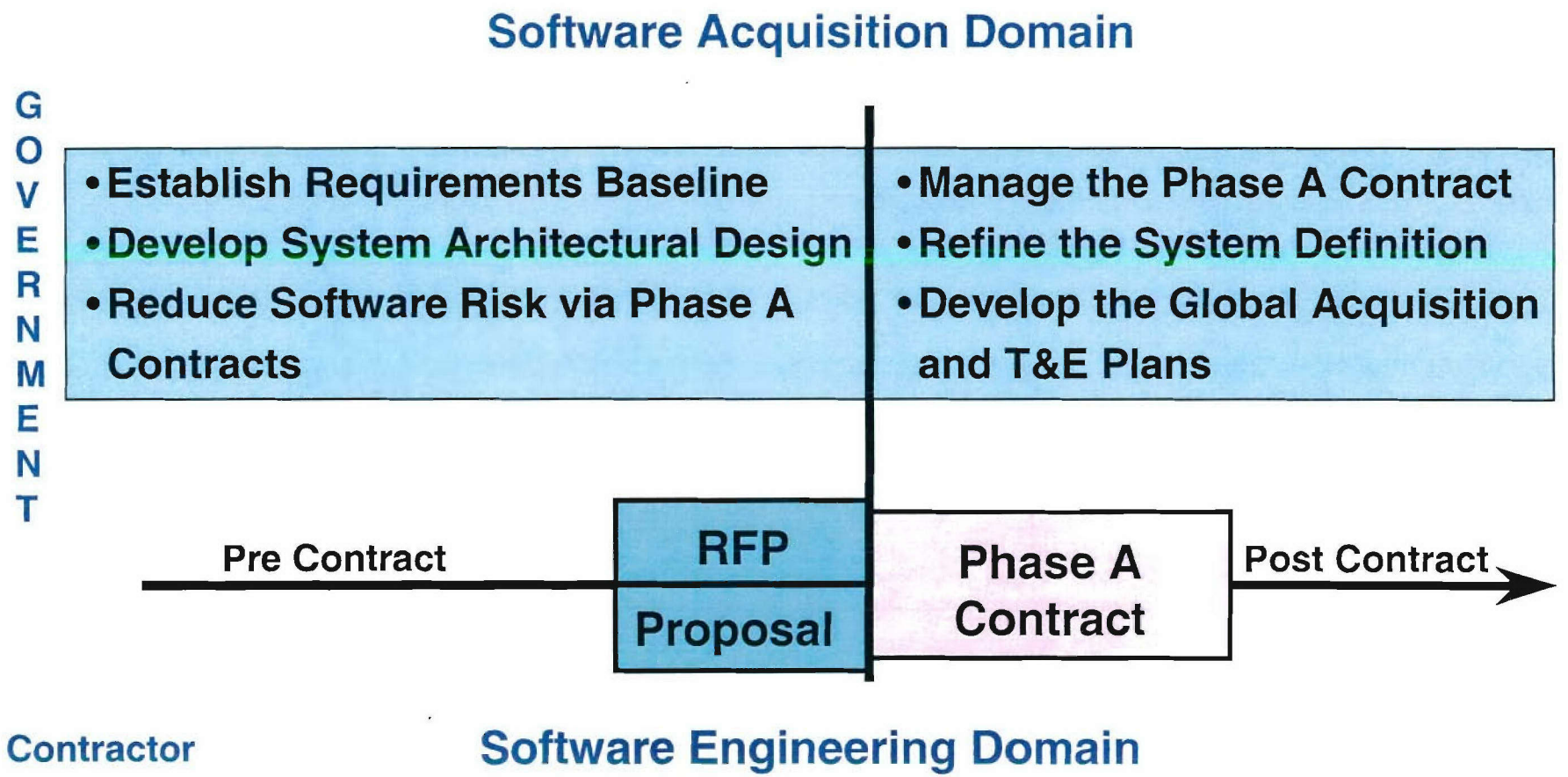


Principal objective of Phase A contract(s)* is to develop the information needed for the government to:

- Solidify the program definition to establish an executable program
  - Includes ensuring the needed technologies are sufficiently mature
- Update the global acquisition strategy, including acquisition plans and products for this and all future evolutions

* Multiple parallel contracts are used frequently in this phase, with selection of a single development contractor for the next phases.

Software Acquisition Best Practices for a Phase A Contract



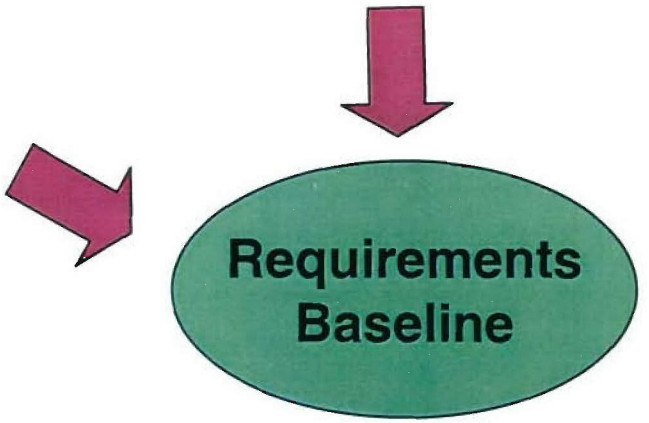
Best Practices for Establishing the Requirements Baseline

Include software in government system performance requirements

- Specialty engineering, especially RMA, safety, security (information assurance), human systems integration, COTS software supportability
- Key Performance Parameters
- Open system architecture and interoperability
- Design for evolution and growth
- Document in Technical Requirements Document (TRD), the Phase A contractual requirements document

Contract for delivery of SW-inclusive reqs. specifications

- Require System and Segment Specifications as CDRL items
- Use System/Subsystem Specification DID (DI-IPSC-81431a)



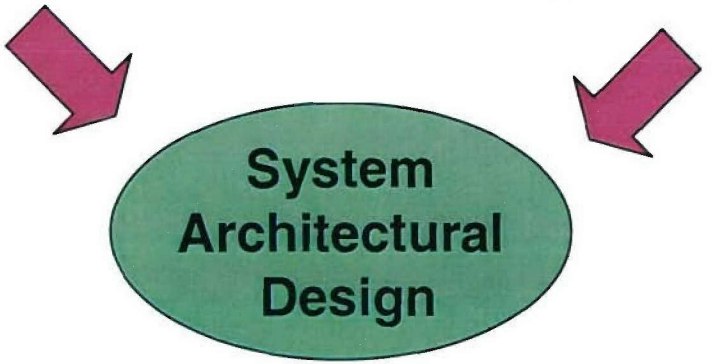
Best Practices for Developing the System Architectural Design

Contract for software architecture trade studies

- With system architecture trades
- Include major software legacy components and COTS software

Contract for delivery of system architecture

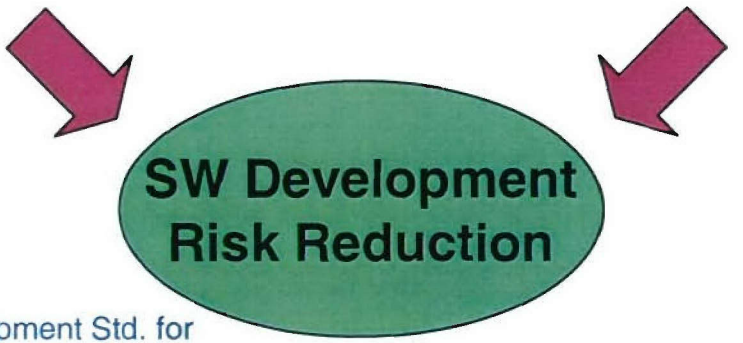
- Require system architecture as a CDRL item
- Require an integrated HW/SW architecture, defined by multiple architecture views
- Include newly developed, reuse and COTS software



Best Practices for Reducing Software Development Risk Via the Phase A Contracts

Contract for software product risk reduction	Contract for software process risk reduction

• Studies/prototyping of high risk areas for software, e.g.,
  • Mission data processing algorithms
  • Mission planning concepts
• Simulation development
• Increase readiness level of computer HW and SW technologies
	• Require delivery of Software Development Plan (DID DI-IPSC-81427a—tailored)
	• Require compliance with robust software development standard*
	• Enable contractor team to prepare for software capability evaluation



* Update to MIL-STD-498: "SW Development Std. for Space Systems," The Aerospace Corporation, TOR-2004(3909)-3537, Rev. A, March 2005.

Best Practices for Managing the Phase A Contracts

Ensure contractor(s) define software-inclusive reqs. specs

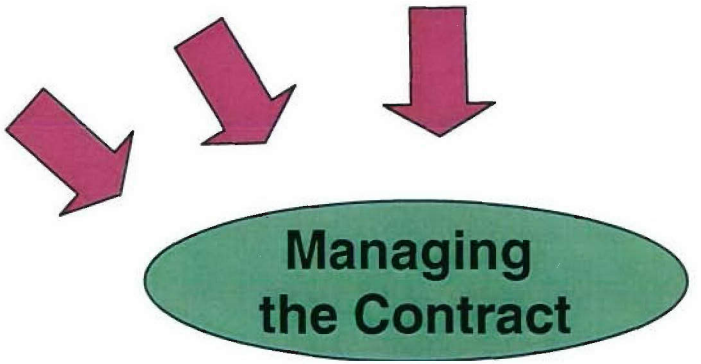
- Software systems engineers (contractor and government) must participate with contractor and gov't. systems engineers

Participate with contractor in software risk reduction

- Government software acquisition personnel with technical expertise in software product and process engineering must participate

Ensure contractor(s) define integrated HW/SW architecture

- Software systems engineers (contractor and government) must participate with contractor and gov't. systems engineers



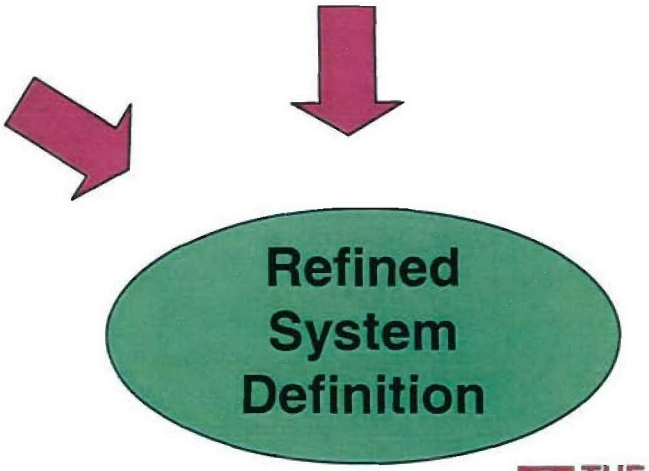
Best Practices for Refining the System Definition

Ensure required capabilities appropriately include software

- Iterate with user to ensure software-related capabilities are appropriately included in the Capability Development Document (CDD), e.g.,
  - Key Performance Parameters
  - RMA, safety, security (information assurance), human systems integration
  - Interoperability
  - Architecture views

Ensure updated CONOPS appropriately includes software

- Iterate with user to ensure any updates to the CONOPS are feasible from a software perspective and consistent with the evolving system architecture and requirements



Best Practices for Developing the Global Acquisition and T&E Plans

Update SW-inclusive program baseline

- Software-inclusive system reqs.
- Integrated HW/SW architecture
- Realistic software size, effort, cost, and schedule estimates for each evolution

Update SW-specific plans

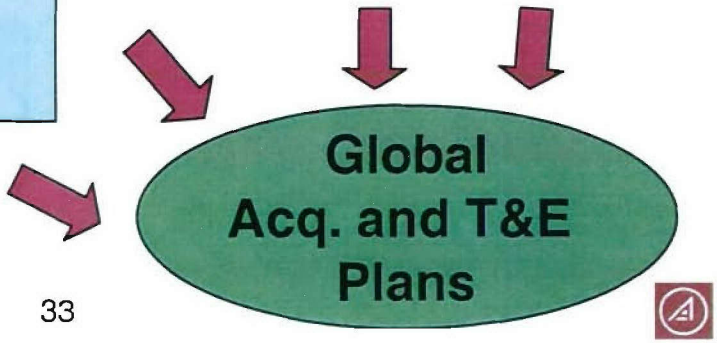
- Software support strategy
- Evaluations of contractor team SW capability
- Software technology insertion
- Software transition to O&M

Update definition of SW-friendly evolutions

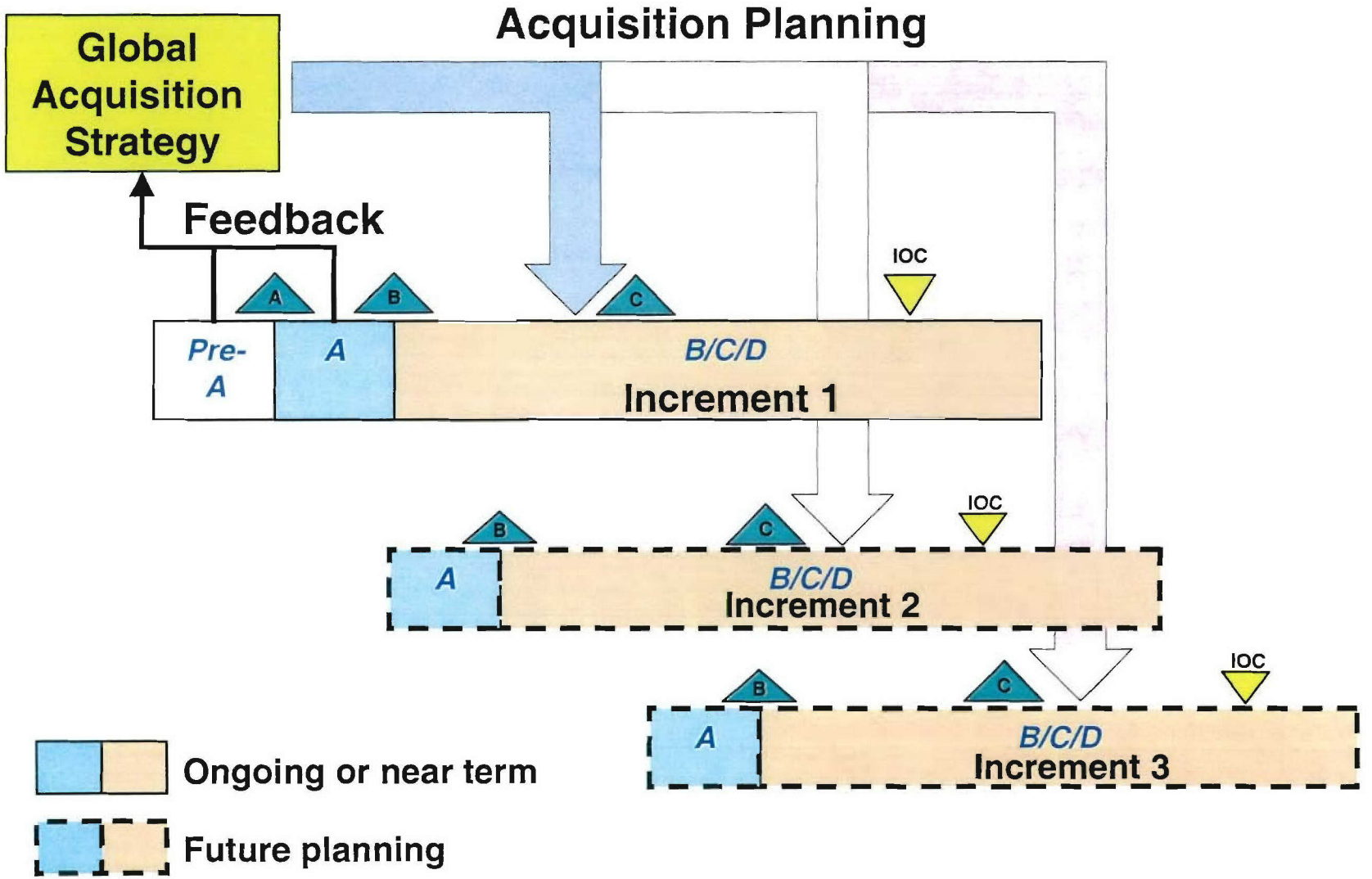
- Evolution capabilities, schedules and integration strategies
- COTS software refresh and legacy software upgrades

Develop SW-inclusive T&E plans

- Part of developing the TEMP



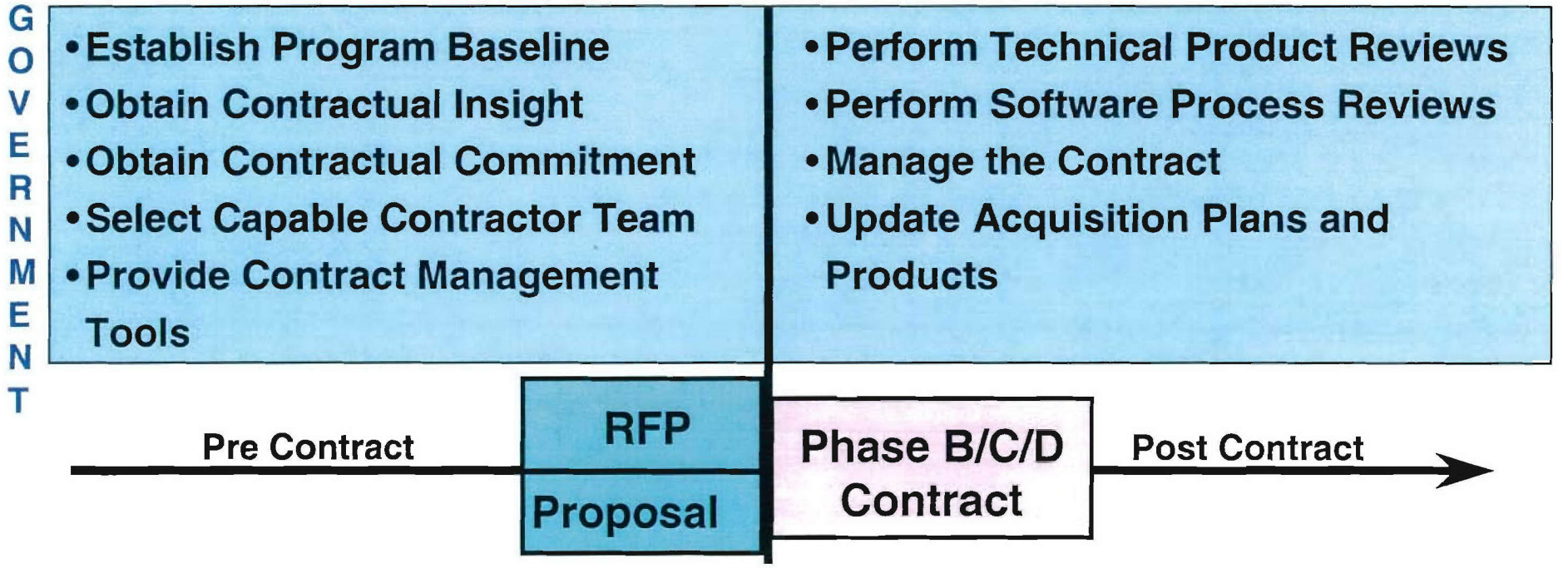
Updating the Global Acquisition Strategy for Evolutionary Acquisition



Phases B/C/D

Software Acquisition Best Practices for a Phase B/C/D Contract

Software Acquisition Domain



Contractor

Software Engineering Domain

Best Practices for Establishing the Program Baseline

Perform software architecture-inclusive trade studies

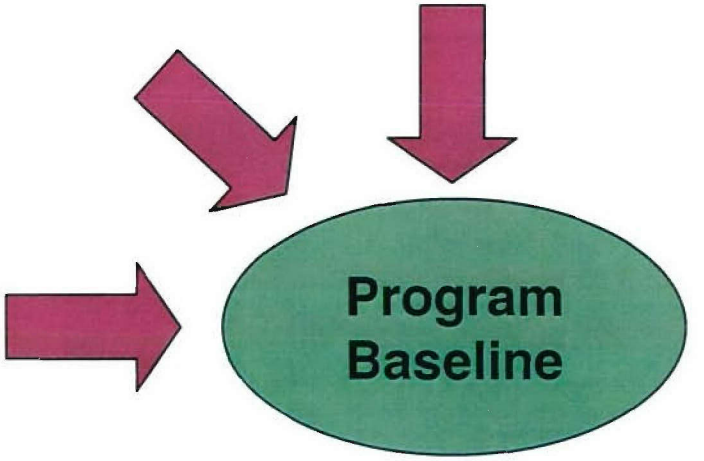
- With system architecture trades
- Include major COTS and legacy components
- Supports government software architecture baseline selection
- Include user in all trades

Include software in system performance requirements

- Specialty engineering, esp. RMA, information assurance, safety
- Key Performance Parameters (KPPs)
- Open system architecture and interoperability
- Prioritized requirements
- COTS software support requirements

Determine realistic, independent baseline software estimates

- Size, effort, cost and schedule
- COTS, reuse and newly developed
- Tasks not reflected in cost models
- Include COTS refresh through both development and sustainment
- Realism especially critical for evolutionary acquisition



Best Practices for Obtaining Contractual Insight

Require key software technical and management deliverables

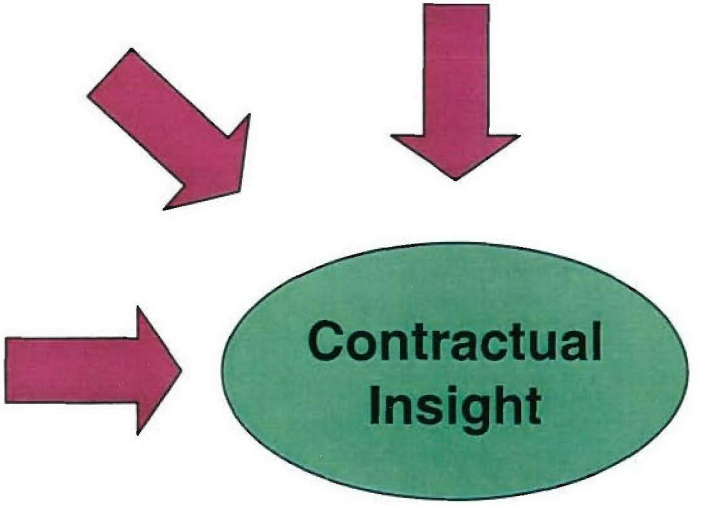
- Highest risk reduction potential:
  - Plans (development, build, transition)
  - Requirements and Architecture
  - Test plans, procedures, and reports
  - Metrics reports
  - Delivery, installation, and maintenance documentation
- Use electronic delivery

Require timely electronic access to all software products

- Intermediate and Final Products
  - Requirements, Architecture, Design
  - Implementation (including code)
  - Integration & Verification Testing
- COTS Evaluation Trade Studies

Require software level technical and management reviews

- In addition to software-inclusive system reviews
- Include COTS software experts in reviews



Recommended Software CDRL Items - 1

- **Plans**
  - Software Development Plan (SDP)
    - Includes the Software Configuration Management Plan, Software Quality Assurance Plan, Software Metrics Plan, Software Reliability Plan
  - Software Master Integration and Verification Plan (Master Build Plan)
  - Software Transition Plan (Transition to O&M)
- **Requirements**
  - Software Requirements Specifications (SRSs)
  - Interface Requirements Specifications (IRSs)
- **Architecture**
  - Software Architecture Description
- **Verification (Qualification Testing)**
  - Software Test Plans
  - Software Test Procedures
  - Software Test Reports
  - Including Software Reliability Test Cases, Reports

Recommended Software CDRL Items - 2

- **Status Reports**
  - Software Metrics Reports
    - Use Practical Software and System Measurement (PSM) resources for guidance
      - Textbook "Practical Software Measurement: Objective Information for Decision Makers"* (especially the Measurement Construct form)
      - Guidebook Practical Software Measurement: A Foundation for Objective Project Management, v. 4.0b1 (see http://www.psmsc.com/PSMGuide.asp)
  - Software Resources Data Report
    - See Defense Acquisition Guidebook, Section 11.3.3, at http://akss.dau.mil/dag/Guidebook/IG_c11.3.3.asp
- **Delivery and Maintenance**
  - Software Version Descriptions
  - Software Product Specifications
- **Operations**
  - Software User's Manual

* McGarry, John, David Card, Cheryl Jones, Beth Layman, Elizabeth Clark, Joseph Dean, and Fred Hall, Practical Software Measurement: Objective Information for Decision Makers, Addison-Wesley, October 2001.

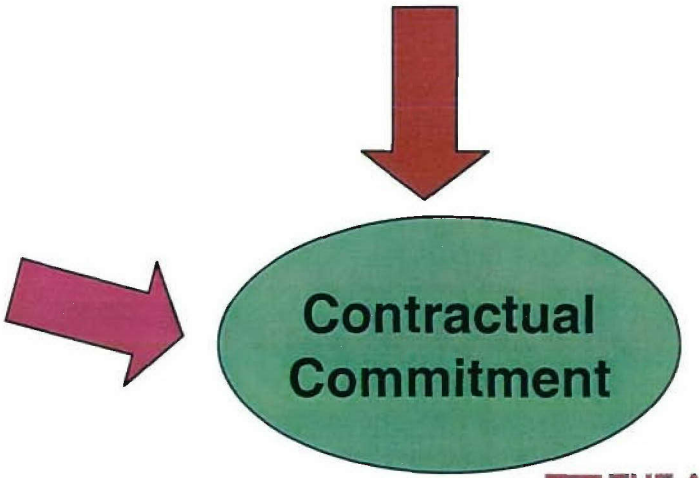
Best Practices for Obtaining Contractual Commitment

Mandate compliance with robust full life cycle SW dev. standard

- TOR-2004(3909)-3537, "SW Development Std. for Space Systems"
  - Full life cycle standard based on MIL-STD-498
  - Updated to bring standard up to date and add mission assurance-related requirements
- Alternatively, use tailored IEEE/EIA J-STD-016
- TOR-2004(3909)-3406, "Recommended Software Stds. for Space Systems" contains other software-related standards recommended for compliance and guidance
  - Also contains J-STD-016 tailoring

Require contractor commitment to Software Development Plan

- Require SDP to include processes for COTS software
- Require Integrated Management Plan (IMP) to have adequate systems engineering & sustainment for COTS
- Include commitment to SDP in IMP



Basic Set of Recommended Software Standards

- Compliance
  - "Software Development Standard for Space Systems," Aerospace TOR-2004(3909)-3537 Rev. A, 11 March 2005
    - Update to MIL-STD-498
  - Software Metrics: "Software Engineering—Software Measurement Process," ISO/IEC 15939
  - Aviation Safety for Software: RCTA/FAA Std DO-178B and RCTA/FAA Std DO-278
  - Software Reliability: "Software Reliability Program Standard" by Society of Automotive Engineers (SAE), SAE JA1002 (2004), January 2004

- Guidance
  - CMMI®-SE/SW/IPPD/SS, V1.1, Staged and Continuous Representations, (SEI-2002-TR-011 and SEI-2002-TR-012), March 2002
  - "IEEE Recommended Practice for Architectural Description of Software Intensive Systems," IEEE 1471
  - PSM textbook and guidebook

Best Practices for Selecting a Capable Software Contractor Team

Evaluate software capability as part of source selection

- Evaluate software capability of offeror teams
  - Individual team member evaluation insufficient
- Evaluate software capability/ processes as subfactor
  - Under Mission Capability factor
  - Weight according to software risk
- Evaluate teams' proposed software processes
  - Corporate and past project process evaluation insufficient

Evaluate software architecture with system design

- Evaluate major HW/SW architecture issues (e.g., space-ground trades, use of COTS & legacy components)

Evaluate realism of cost and schedule bids

- Suspect extremes of productivity, COTS & reuse, & low lines of code
- Ensure all COTS SW tasks included
- Ensure bids contain enough margin

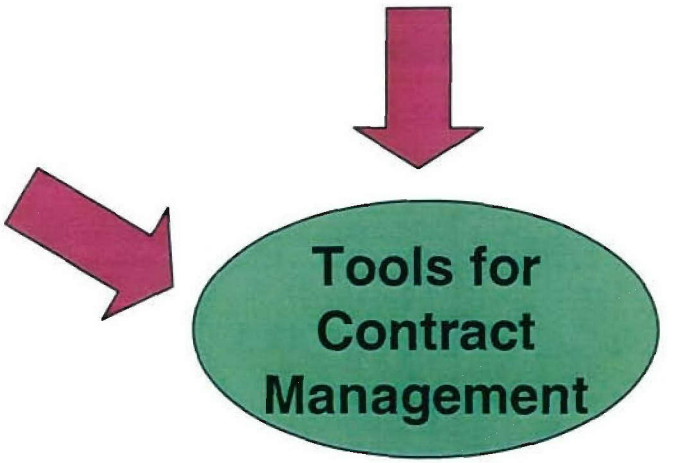
Best Practices for Providing Tools for Contract Management

Incentivize software quality,* not just cost and schedule

- Use award and incentive fee plans
- Reward adherence to
  - Defined software processes
  - Software process improvement
- Reward timely and adequate response to government comments
- Reward low rework rates
- Reward meeting RMA requirements post delivery/launch
- Reward architecture development that supports COTS software evolution and legacy transition

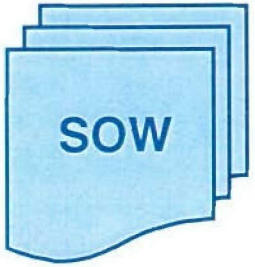
Mandate periodic team software capability appraisals

- Relate results and improvement actions directly to award fee
- Explicitly include COTS processes in appraisals



* Quality in this context is producing work products that do not require rework in successor activities.

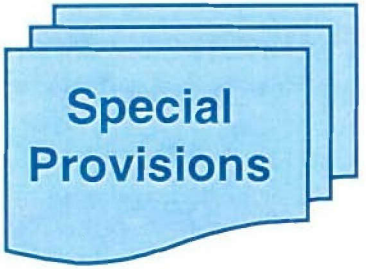
Software Acquisition "Best Practice" Contract



- Comply with SDP
- Do COTS SW trade studies
- Hold SW technical reviews
- Undergo periodic software process appraisals

- Software-inclusive system requirements
- COTS software support requirements

- Software plans
- Reqs and architecture
- Test documentation
- Metrics reports
- O&M documentation



- Electronic access to all software products
- Access to prime and subcontractor software technical and mgmt data

- Full life cycle software standard
- Other software-related standards

- Software quality
- COTS architecture evolution and legacy transition

Some Acquisition Do's and Don'ts for COTS-Based Software Systems

- **DON'T require the following: "Maximize the use of COTS software"**
  - **DO** require a balanced solution among newly developed, reuse and COTS software that meets cost, schedule and performance objectives
- **DON'T force the developers to commit to their COTS software selections before they have had time to do thorough evaluations**
  - **DO** allow flexibility in schedules for software reviews and deliverables
- **DON'T force the use of a life cycle model that must define all software requirements up front**
  - **DO** provide flexibility to use evolutionary or spiral life cycle models
- **DON'T blame the contractors for COTS-software related events that are beyond their control**
  - **DO** reward mitigating the effects of unforeseen COTS problems
- **DON'T use commercial item procurements for large, complex ground or space systems**
  - **DO** use contracts that require full application of systems and software engineering disciplines
- **AVOID modifying COTS software, except as a last resort**

Best Practices for Performing Technical Product Reviews

Perform in-depth technical reviews of software products

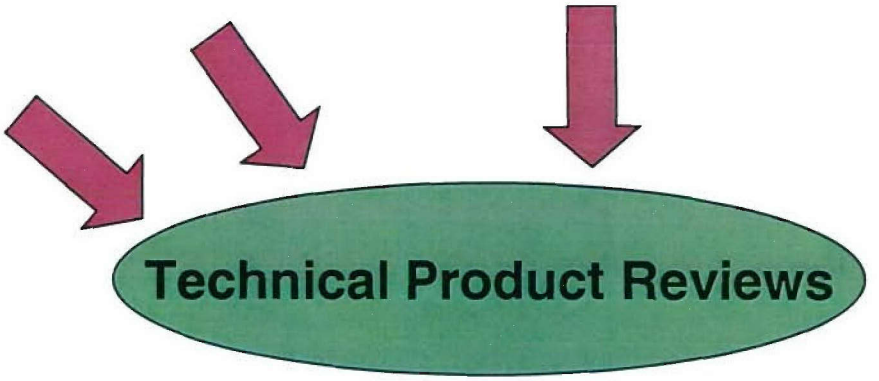
- IPTs, TIMs, working groups, peer reviews, etc.
- Software Level Technical Reviews
- High risk/critical software products
- Key software technical deliverables
- Focus on areas of highest risk

Monitor software integration and verification adequacy

- Begin at the build level
- Focus on areas of highest risk
- Focus on early performance analysis results and meeting KPPs
  - Ensure COTS SW performance is measured
- Ensure requirements allocated to COTS software are verified

Include users/operators in all technical review activities

- Focus on operational suitability of evolving software-intensive system
- Including COTS software capabilities and impacts on O&M



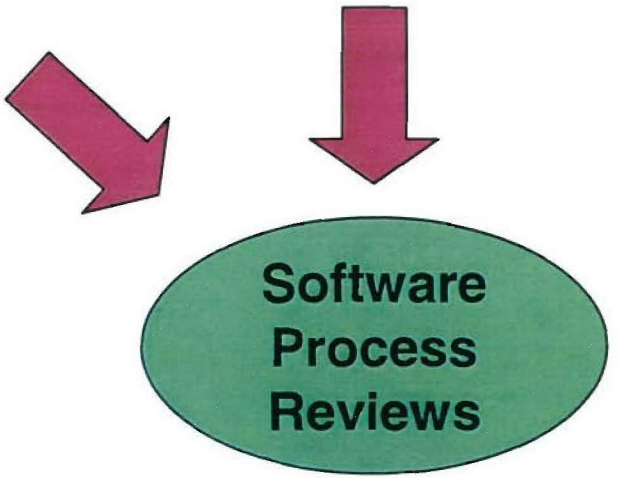
Best Practices for Performing Software Process Reviews

Review effectiveness of contractor team's SW processes

- Review team's adherence to defined software processes
  - Identify adherence deficiencies
  - Assist in deficiency correction
- Evaluate effectiveness of defined SW processes
  - Identify process deficiencies
  - Assist with process improvement
- Level 2 and 3 CMMI®/SW-CMM® adherence for an individual team member may not be sufficient*
  - Need to address full team

Perform periodic team software capability appraisals

- During contract performance
- Support for significant program or award fee milestones
- Explicitly include COTS software processes



* CMM is registered in the U.S. Patent Office and Trademark Office by Carnegie Mellon University.

Best Practices for Managing the Development Contract

Use incentive/award fees aggressively

- Motivate good software practices
- Focus on quality and architecture

Ensure satisfaction of software –inclusive requirements

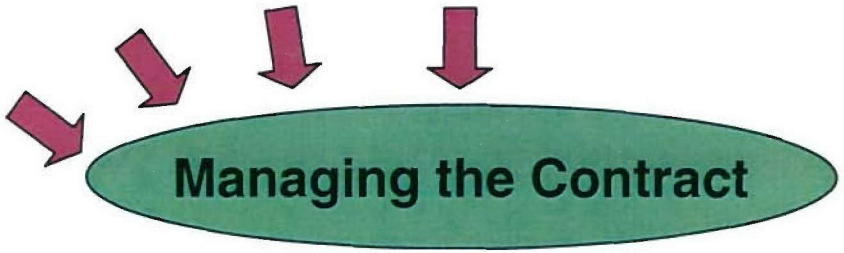
- Especially RMA, info. assurance, safety, COTS supportability

Apply proactive quantitative management

- Ensure a comprehensive software/ system metrics program balanced across information categories
  - Include leading quality indicators
- Perform cross-metric analysis
- Earned value at the software level is necessary; but earned value alone is not sufficient

Perform periodic independent assessments

- Support for significant program or award fee milestones
- Act aggressively on findings



Basic Recommended Software Metrics - 1

- **Product Size**
  - Planned vs. actual for reuse, modified reuse and newly developed software
- **Staffing**
  - Planned vs. actual
- **Requirements Volatility**
  - Total number and numbers added, deleted and modified
- **Requirements Verification Status**
  - Planned vs. actual number verified
- **Problem/Change Report Status**
  - Total number opened and closed, number opened and closed during reporting period; and by age and severity
- **Build Contents**
  - Planned vs. actual requirements and number of units
- **Software Earned Value**
  - ACWP, BCWP, BCWS, SPI, CPI at the software WBS levels

Basic Recommended Software Metrics - 2

- **Milestone Status**
  - Planned vs. actual dates; number of days slipped
- **Effort**
  - Planned vs. actual for total and by function
- **Rework**
  - Amount of effort expended to revise products due to problems or changes
- **Progress**
  - Planned vs. actual units designed, implemented, tested, integrated
  - Planned vs. actual integration and qualification test procedures completed and dry run, tests successfully executed
- **Computer Hardware Resource Utilization**
  - Planned vs. actual utilization for processors, memory, storage, networks, I/O channels and busses

Best Practices for Updating Acquisition Plans and Products

Ensure updated CDD and CONOPS are software-inclusive

- Iterate with user to ensure software is still appropriately reflected in their updated CDD and CONOPS

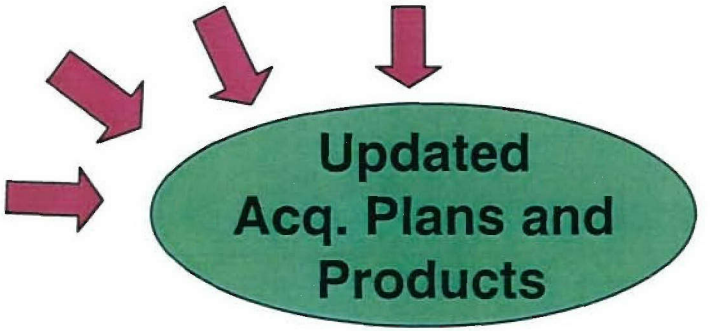
Update SW-inclusive program baseline

- Software-inclusive system reqs.
- Integrated HW/SW architecture
- Realistic software size, effort, cost, and schedule estimates for each evolution

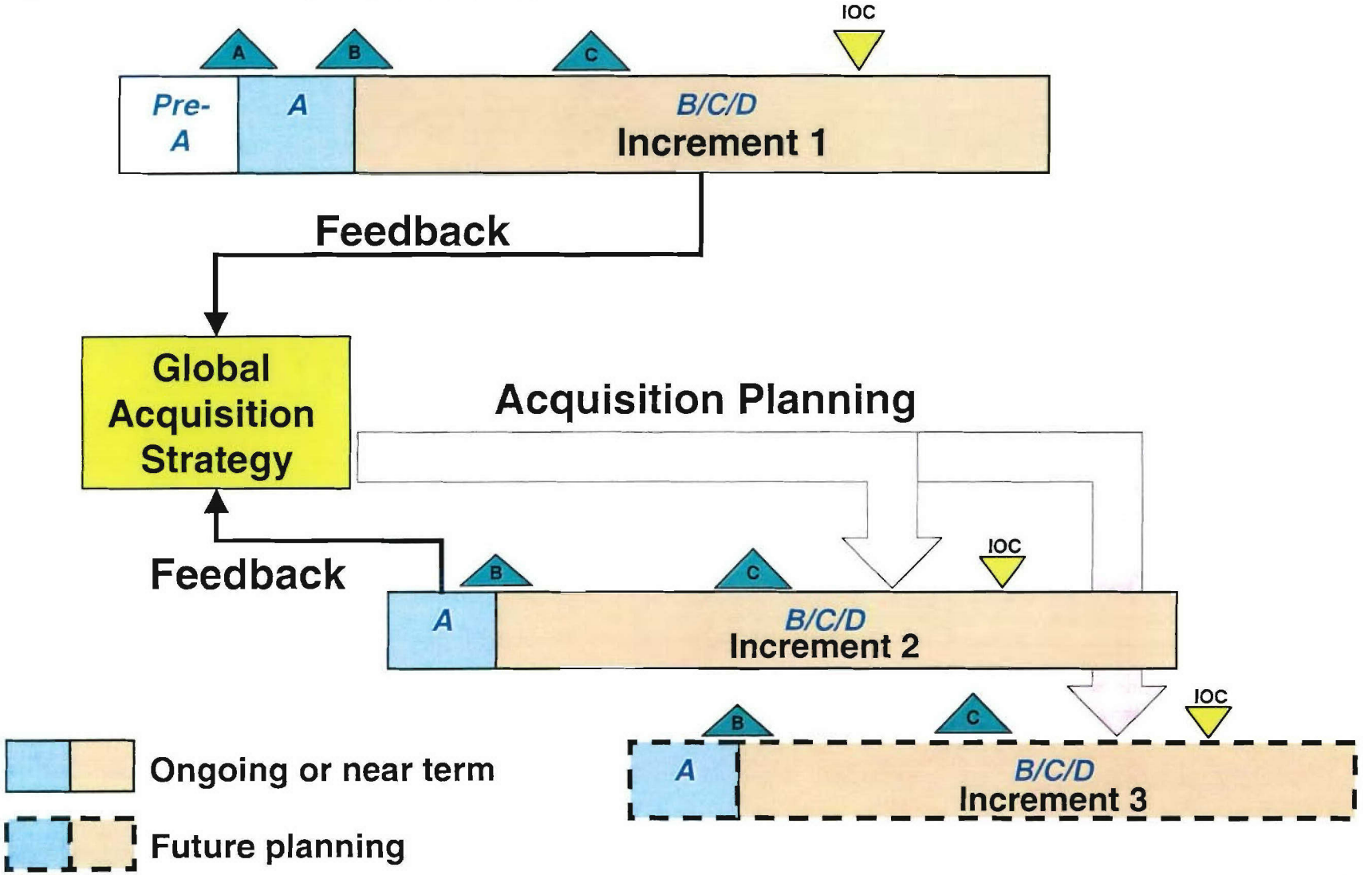
Update SW-inclusive TEMP

Update SW-inclusive acquisition plans

- Evolution capabilities, schedules and integration plans
- Plans for COTS software refresh and legacy software upgrades
- Software support plans
- Plans for evaluations of contractor team SW capability
- Plans for software technology insertion
- Software transition to O&M



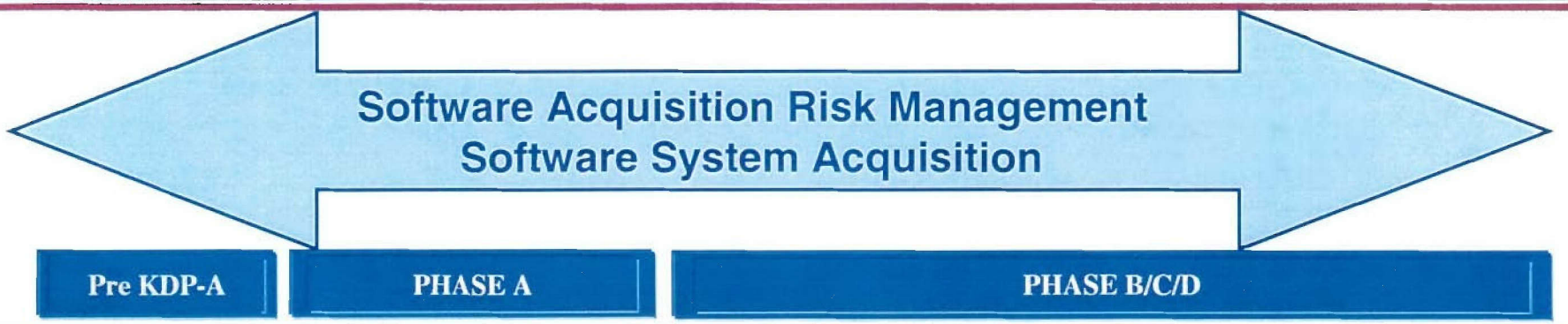
Updating the Global Acquisition Strategy for Evolutionary Acquisition - Revisited



This page is intentionally blank

Full Acquisition Life Cycle

Best Practices that Span the DOD and NSS Acquisition Life Cycle



Software Acquisition Risk Management

- Continuous software acquisition risk management
  - Across the entire acq. life cycle
  - Across all evolutions
  - Within each ongoing evolution
- Program level risk management and contractor development risk management are necessary but not sufficient
- Ensure management reserves are consistent with SW risks

Software System Acquisition

- Integrate software acquisition with the system acquisition process
  - From capability needs identification through system retirement
  - Especially during early life cycle and pre-contract award activities

Outline

- **Definitions and Background**
  - Software Acquisition vs. Software Engineering
  - Best Practices
- **Software Acquisition Best Practices**
  - Pre-Phase A
  - Phase A
  - Phases B/C/D
  - Full Acquisition Life Cycle
- **Software Acquisition Best Practices from Other Organizations**
  - Data and Analysis Center for Software (DACS)
  - Software Engineering Institute (SEI)
- **Current Air Force and SMC Software Acquisition Policy**
- **Conclusion**
- **Acronym List**

This page is intentionally blank

Data and Analysis Center for Software (DACS)

- **The Data and Analysis Center for Software (DACS) is a Department of Defense (DOD) Information Analysis Center (IAC)**
  - ❖ Designated as the DOD Software Information Clearing House serving as an authoritative source for state-of-the-art software information and providing technical support for the software community

- **The DACS is administratively managed by the Defense Technical Information Center (DTIC) under the DOD IAC Program**

- **The DACS is technically managed by Air Force Research Laboratory - Information Directorate (AFRL/IF)**

- **DACS home page is at http://www.thedacs.com**

DACS Gold Practice Initiative

- **The DACS has established a "Gold Practices" initiative to provide information about prevalent software acquisition and development "best" practices that may have a positive impact on program risks and return on investment**

- **Information can be found on the Gold Practices website http://www.goldpractices.com**
  - ❖ Sign up to become a member on this website

- **Information on the Gold Practices website is continually being updated**
  - ❖ Information on 15 "Gold Practices" is currently available

The 15 Current Gold Practices - 1

Gold Practice	Development?	Acquisition?
1. Acquisition Process Improvement	No	Yes
2. Architecture-First Approach	Yes	Yes*
3. Assess Reuse Risks and Costs	Yes	Yes
4. Binary Quality Gates at the Inch-Pebble Level	Yes	No
5. Commercial Specifications/Open Standards	Yes	Yes*
6. Ensure Interoperability	Yes	Yes*
7. Formal Inspections	Yes	No
8. Formal Risk Management	Yes	Yes

* Can be adapted to apply to acquisition

The 15 Current Gold Practices - 2

Gold Practice	Development?	Acquisition?
9. Integrated Product and Process Development (IPPD)	Yes	Yes*
10. Manage Requirements	Yes	Yes*
11. Model-Based Testing	Yes	No
12. Plan for Technology Insertion	Yes	Yes*
13. Requirements Trade-Offs/ Negotiations	Yes	Yes*
14. Statistical Process Control	Yes	No
15. Track Earned Value	Yes	Yes*

* Can be adapted to apply to acquisition

Software Engineering Institute (SEI)

- **The Software Engineering Institute (SEI) is a Federally Funded Research and Development Center (FFRDC) whose focus is to improve the state of software practice throughout the defense community**
- **The SEI has an extensive website, with home page http://www.sei.cmu.edu**
- **The SEI has a number of initiatives, one of which is in Acquisition**
  - ❖ See http://www.sei.cmu.edu/acquisition/acquisition.html

The Software Acquisition Capability Maturity Model®

- **The Software Acquisition Capability Maturity Model® (SA-CMM®) was developed by the SEI with extensive participation of the DOD software acquisition community***
  - First published in 1999
  - Latest version (1.03) published in 2003
  - See http://www.sei.cmu.edu/arm/SA-CMM.html
- **The SA-CMM® contains 17 Key Process Areas, each containing a set of related Best Practices**
- **Has been successfully used by numerous acquisition organizations as the basis for software acquisition process improvement**
- **The SA-CMM® is no longer being maintained, but still contains valuable guidance**

* Capability Maturity Model is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

SA-CMM® Key Process Areas

LEVEL	FOCUS	KEY PROCESS AREAS
5 Optimizing	Continuous process improvement	• Acquisition Innovation Management • Continuous Process Improvement
4 Quantitative	Quantitative management	• Quantitative Acquisition Management • Quantitative Process Management
3 Defined	Process Standardization	• Training Program • Acquisition Risk Management • Contract Performance Management • Project Performance Management • User Requirements • Process Definition and Maintenance
2 Repeatable	Basic project management	• Transition to Support • Evaluation • Contract Tracking and Oversight • Project Management • Requirements Development & Management • Solicitation • Software Acquisition Planning
1 Initial	Competent people and heroics	

The CMMI® Acquisition Module

- **Recently the SEI prepared an Acquisition Module for the CMMI® (CMMI®-AM)**
  - ❖ Focuses on effective acquisition activities and practices that are implemented by first-level acquisition projects (e.g., SPOs)
  - ❖ Not software-specific, but can be applied to software acquisition
  - ❖ Published in 2004, the CMMI®-AM, Version 1.1, is available at http://www.sei.cmu.edu/publications/documents/05.reports/05tr011.html

- **The CMMI®-AM, Version 1.1, contains 12 Process Areas, each containing a set of related best practices**

CMMI®-AM, Version 1.1, Process Areas

- **Project Management Process Areas**
  - Project Planning
  - Project Monitoring and Control
  - Solicitation and Contract Monitoring
  - Integrated Project Management
  - Risk Management
- **Engineering Process Areas**
  - Requirements Development
  - Requirements Management
  - Verification
  - Validation
- **Support Process Areas**
  - Decision Analysis and Resolution
  - Measurement and Analysis
  - Transition to Operations and Support

This page is intentionally blank

Outline

- **Definitions and Background**
  - Software Acquisition vs. Software Engineering
  - Best Practices
- **Software Acquisition Best Practices**
  - Pre-Phase A
  - Phase A
  - Phases B/C/D
  - Full Acquisition Life Cycle
- **Software Acquisition Best Practices from Other Organizations**
  - Data and Analysis Center for Software (DACS)
  - Software Engineering Institute (SEI)
- **Current Air Force and SMC Software Acquisition Policy**
- **Conclusion**
- **Acronym List**

Air Force Software Acquisition Policy - 1

- **"Revitalizing the Software Aspects of Systems Engineering," 04A-003, 20 September 2004**
  - Signed by Dr. Sambur (SAF/AQ) and Mr. Teets (SAF/US)
- **Implements Section 804 of the National Defense Authorization Act of FY 2003 for the Air Force**
  - Section 804 requires the implementation of a software acquisition process improvement program
- **Available from SAF/AQ website: https://www.safaq.hq.af.mil/mil/organizations_mil/ace_mil/documents/Sofware Aspects of Sys Eng.pdf**

Air Force Software Acquisition Policy - 2

- **Topics addressed in the policy**
  - High Confidence Estimates
  - Realistic Program Baselines
  - Risk Management
  - Capable Developer
  - Developer Processes
  - Program Office Processes
  - Earned Value Applied to Software
  - Metrics
    - Size, Effort, Schedule, Requirements Definition and Stability, Staffing, Progress (Design, Code, Test), Computer Resource Utilization
  - Life Cycle Support
  - Lessons Learned

SMC Software Acquisition Policy and Instructions - 1

- **Policy on Software Acquisition at SMC, 20 August 2004 (SMC/CC) – signed by Lt Gen Arnold**
  1. Perform a software capability appraisal using either the SDCE or a CMMI®-based appraisal
  2. Use a software development standard from the SMC approved standards list
  3. Follow the SMC Software Acquisition Process Improvement Instruction
     - "Software Acquisition Process Improvement Instruction", AFI63-103, 28 March 2005
     - Implements Air Force policy and Section 804 at SMC
     - Available from AF e-publishing or SMC publications website
       - SMC Publications (Click on SMC publications; then scroll down to 63-103) http://intranet.losangeles.af.mil/61ABG/61CS/SMCPDL

SMC Software Acquisition Policy and Instructions - 2

- **SMC policy (continued)**
  4. Use the SMC Software Acquisition Handbook for guidance
     - "Software Acquisition Handbook", Version 2.0
     - Available on the SMC/AXE website (Scroll down and select Software Acquisition Handbook in left column; then click SAH):
       - http://ax.losangeles.af.mil/axe/ax_workspace/ax/axe/index.php
- **New SMC Software Acquisition Instruction soon to be published**
  - ❖ Has been through SPO reviews; in final approval and publication cycle
  - ❖ Covers all software acquisition requirements for SMC programs

This page is intentionally blank

Outline

- **Definitions and Background**
  - Software Acquisition vs. Software Engineering
  - Best Practices
- **Software Acquisition Best Practices**
  - Pre-KDP A
  - Phase A
  - Phases B/C/D
  - Full Acquisition Life Cycle
- **Software Acquisition Best Practices from Other Organizations**
  - Data and Analysis Center for Software (DACS)
  - Software Engineering Institute (SEI)
- **Current Air Force and SMC Software Acquisition Policy**
- **Conclusion**
- **Acronym List**

This page is intentionally blank

Conclusion

- **Software acquisition Best Practices do not guarantee success**
  - ❖ They are not a panacea!
- **Using Best Practices, however, can reduce risk in complex software-intensive system acquisitions**
- **Recent Air Force and SMC policy requires implementation of improved software acquisition processes and use of Best Practices**

This page is intentionally blank

Outline

- **Definitions and Background**
  - Software Acquisition vs. Software Engineering
  - Best Practices
- **Software Acquisition Best Practices**
  - Pre-Phase A
  - Phase A
  - Phases B/C/D
  - Full Acquisition Life Cycle
- **Software Acquisition Best Practices from Other Organizations**
  - Data and Analysis Center for Software (DACS)
  - Software Engineering Institute (SEI)
- **Conclusion**
- **Acronym List**

This page is intentionally blank

Acronyms and Abbreviations - 1

Acq	Acquisition
ACWP	Actual Cost of Work Performed
AF	Air Force
AFI	Air Force Instruction
AFMCP	Air Force Materiel Command Pamphlet
AFRL/IF	Air Force Research Laboratory-Information Directorate
AM	Acquisition Module
BCWP	Budgeted Cost of Work Performed
BCWS	Budgeted Cost of Work Scheduled
CBSS	COTS-Based Software System
CDD	Capability Development Document
CDR	Critical Design Review
CDRL	Contract Data Requirements List
CMM®	Capability Maturity Model®
CMMI®	Capability Maturity Model® Integration℠ *
CMMI®-AM	Capability Maturity Model® Integration℠ – Acquisition Module
CMU	Carnegie Mellon University
CONOPS	Concept of Operations
COTS	Commercial Off the Shelf
CPI	Cost Performance Index
DACS	Data and Analysis Center for Software
DID	Data Item Description

Acronyms and Abbreviations - 2

DOD	Department of Defense
DODI	DoD Instruction
DTIC	Defense Technical Information Center
EIA	Electronic Industries Alliance
Eng	Engineering
FFRDC	Federally Funded Research and Development Center
FOC	Full Operational Capability
FY	Fiscal Year
HW	Hardware
I/O	Input/Output
IAC	Information Analysis Center
ICD	Initial Capabilities Document
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IMP	Integrated Management Plan
IPPD	Integrated Product and Process Development
IPT	Integrated Product Team
IOC	Interim Operational Capability
IRS	Interface Requirements Specification
ISO	International Organization for Standardization
J	Joint
KDP	Key Decision Point

Acronyms and Abbreviations - 3

KPP	Key Performance Parameter
MIL	Military
MOIE	Mission-Oriented Investigation and Experimentation
MS	Milestone
NSS	National Security Space
O&M	Operations and Maintenance
PDR	Preliminary Design Review
PSM	Practical Software and System Measurement
RFP	Request for Proposal
RMA	Reliability, Maintainability, Availability
SA	Software Acquisition
SA-CMM®	Software Acquisition Capability Maturity Model®
SCAMPI℠	Standard CMMI®-Based Appraisal Method for Process Improvement
SE	Systems Engineering
SEI	Software Engineering Institute
SDCE	Software Development Capability Evaluation
SDP	Software Development Plan
SDR	System Design Review
SFR	System Functional Review
SM	Service Mark
SMC	Space and Missile Systems Center

Acronyms and Abbreviations - 4

SOW	Statement of Work
SPI	Schedule Performance Index
SPO	System Program Office
SRR	System Requirements Review
SRS	Software Requirements Specification
SS	Supplier Sourcing
STD	Standard
SW	Software
SW-CMM®	Capability Maturity Model® for Software
T&E	Test and Evaluation
TEMP	Test and Evaluation Master Plan
TIM	Technical Interchange Meeting
TOR	Technical Operating Report
TR	Technical Report
TRD	Technical Requirements Document
USAF	United States Air Force
vs.	Versus
WBS	Work Breakdown Structure

THE AEROSPACE CORPORATION
2350 E. El Segundo Boulevard
El Segundo, California 90245-4691
U.S.A.